(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号

特開平8－77025

(43)公開日　平成８年(1996)３月22日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　9/46	３４０　Ｃ	7737－5B		
	３２２　Ｃ	7737－5B		

審査請求　未請求　請求項の数２　ＯＬ　（全５頁）

(21)出願番号　特願平6－208746

(22)出願日　平成６年(1994)９月１日

(71)出願人　000001122
国際電気株式会社
東京都中野区東中野三丁目14番20号

(72)発明者　堀井　貞義
東京都中野区東中野三丁目14番20号　国際電気株式会社内

(72)発明者　保井　毅
東京都中野区東中野三丁目14番20号　国際電気株式会社内

(72)発明者　布村　一朗
東京都中野区東中野三丁目14番20号　国際電気株式会社内

(74)代理人　弁理士　高崎　芳紘

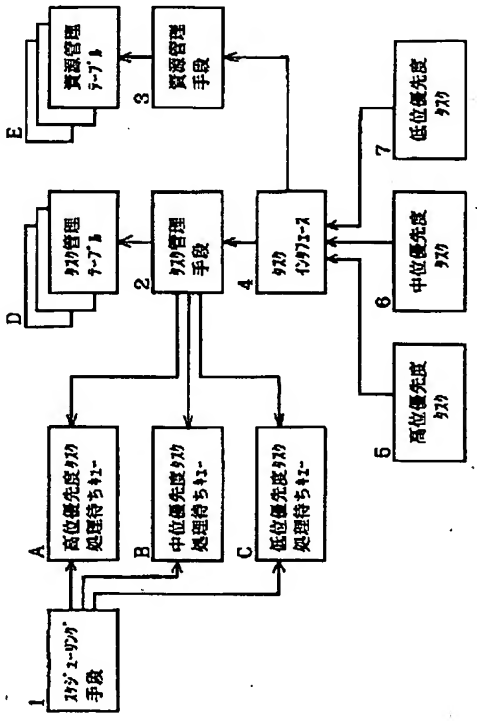
(54)【発明の名称】　タスクの優先度制御方法、タスクの優先度制御装置

(57)【要約】

【目的】　オーバヘッドの増加を極力抑えてタスク優先度の逆転現象を防止する。

【構成】　低位優先度のタスク７が資源管理手段３を介して共有資源を専有使用している最中に、高位優先度のタスク５から同じ共有資源の使用要求が発行されると、タスク管理手段２はタスク７の優先度を共有資源の使用終了までの間、タスク５の優先度に一時変更する。

【効果】　優先度の一時変更により、タスク７の共有資源使用中に中位優先度のタスク６から割り込みが入ってタスク７の処理が後回しになり、結果としてタスク６がより優先度の高いタスク５より先に処理を行うという、逆転現象が防止される。



1

【特許請求の範囲】

【請求項１】　複数のタスクの各々にその優先度を予め設定し、より優先度の高いタスクを優先して実行するシステムのタスクの優先度制御方法において、

１つのタスクがある共有資源を占有して使用している最中に、より優先度の高い別のタスクから上記共有資源の使用要求が発行されたとき、

上記１つのタスクの優先度を上記別のタスクの優先度に一時変更し、上記１つのタスクによる上記共有資源の使用が終了したとき、上記一時変更した優先度を元の優先度へ戻すことを特徴とするタスクの優先度制御方法。

【請求項２】　複数のタスクに優先度を設定する第１の手段と、

１つのタスクがある共有資源を占有しているときに別のタスクが上記共有資源の使用要求を発行したときに、上記共有資源を占有しているタスクの優先度を検出する第２の手段と、

該第２の手段により検出した優先度が上記別のタスクの優先度より低いときに上記１つのタスクの優先度を上記別のタスクの優先度に一時変更する第３の手段と、上記１つのタスクによる上記共有資源の占有使用が終了したときに上記一時変更した１つのタスクの優先度を元へ戻す第４の手段とを備えたことを特徴とするタスクの優先度制御装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、タスクの優先度制御方法およびその装置に係わり、とくに半導体制御装置などのリアルタイム性が要求される制御システムに適したタスクの優先度制御方法に関する。

【０００２】

【従来の技術】半導体制御装置などの従来のリアルタイムオペレーティングシステムでは、オペレーティングシステムのリアルタイム性を補償する手段として、固定優先度スケジューリングをサポートしている。すなわち、複数のタスクの各々に予め優先度を定めておいて、高い優先度のタスクを優先して実行するようにしている。

【０００３】また、オペレーティングシステム資源へのアクセスを割り込み可能（Preemption）にする試みも行われている。例えば、Unix System V R4.0, Lynxos などがあり、これらは、オペレーティングシステムの処理を細かく分析し、中断可能なコード部分を見つけてそこで割り込みを行えるようにすることで、オペレーティングシステムの処理中であっても割り込みによりオペレーティングシステム資源を優先度のより高いタスクが優先使用できるようにするものである。

【０００４】さらに、あるタスクが共有資源を使用して処理を行っているときに、より優先度の高いタスクからの処理要求が発生したとき、その高い優先度を一時的に実行中のタスクに継承させる方式が、カーネギメロン大学の ARTS (Advanced Real Time System) や、Real-Time Mach として研究されている。

2

【０００５】

【発明が解決しようとする課題】固定優先度スケジューリングを行う場合、資源利用の競合が生じたときに、優先度の低いタスクが高いタスクより先に実行されてしまうという、優先度の逆転現象が発生することがある。図４はその現象の説明図で、低い優先度のタスク１がある資源Ａをロックしているとする（状態Ｓ１）。このとき高い優先度のタスク４が起動され、同じ資源Ａを要求したとすると、資源Ａはすでにタスク１にロックされているため、その解除待ちの状態Ｓ２となる。そしてタスク１の処理が続けられる（状態Ｓ３）。このときタスク１のロックが解除される前に、中位の優先度のタスク２あるいはタスク３が起動されてその処理状態Ｓ３、Ｓ４などになると（タスク２、３から資源Ａは要求されていないとする）、そのあいだ低い優先度のタスク１は資源Ａをロックしたまま待ち状態になる。中位の優先度のタスク２または３などの処理が終わってタスク１の処理状態に戻り、さらにこれが終わって資源Ａのロックが解除され、ここで初めてタスク４による資源Ａのロック（状態Ｓ７）が可能になる。こうしてタスク４より低い優先度のタスク２、３などが、タスク４より先に実行されてしまうという、優先度の逆転現象が生じ、リアルタイム性を損なうことになる。

【０００６】前記従来技術の内、Unix System V R4.0 などの方法では、オペレーティングシステム資源へのアクセスを割り込み可能にすることはできても、ユーザレベルで使用する資源に関しては関知していない。これは、ユーザレベルで資源の排他を制御しているものに関しては、オペレーティングシステムは関与できないからである。

【０００７】また、カーネギメロン大学で研究中の ARTS などの方法では、オペレーティングシステムはタスクの優先度構成に係わらず、すべての資源の排他制御において、優先度の逆転現象を解消しようと、タスクが排他資源を要求する時点すべてで優先度継承のための処理を行う。しかし、同じ資源を利用するタスクの間に優先度の差異がない場合、あるいは同じ資源を利用する高位の優先度のタスクと、低位の優先度のタスクの間の優先度を持つタスクが存在しない場合には、優先度の継承処理は不要であり、このためシステムのオーバヘッドは増大する。

【０００８】本発明の目的は、システムのオーバヘッド無しに優先度の逆転現象を防止することのできる、タスクの優先度制御方法およびその装置を提供するにある。

【０００９】

【課題を解決するための手段】上記の目的は、複数のタスクの各々にその優先度を予め設定し、より優先度の高いタスクを優先して実行するシステムのタスクの優先度

制御方法において、１つのタスクがある共有資源を占有して使用している最中に、より優先度の高い別のタスクから上記共有資源の使用要求が発行されたとき、上記１つのタスクの優先度を上記別のタスクの優先度に一時変更し、上記１つのタスクによる上記共有資源の使用が終了したとき、上記一時変更した優先度を元の優先度へ戻すことにより達成される。

【００１０】

【作用】１つのタスクがある共有資源を占有している最中に、より優先度の高い別のタスクがその共有資源の使用要求をしたとき、占有中のタスクの優先度を別のタスクの優先度に一時置き換えるから、上記１つのタスクの優先度と別のタスクの優先度の中間の優先度を持つ第３のタスクは割り込めないから、上記１つのタスクの実行を遅らせ、従って上記別のタスクの上記共有資源へのアクセスを遅らせるという、優先度の逆転現象を防止できる。また、優先度の一時変更は、より優先度の高いタスクからの共有資源要求がその共有資源が他のより低い優先度のタスクに占有されているときだけ行われるので、優先度の一時変更のためのオーバヘッド増加は必用最小限に抑えられる。

【００１１】

【実施例】以下、本発明を実施例により詳細に説明する。図３は、タスクの優先度にもとづいてマルチタスクの制御を行うシステムの機能ブロック図で、処理待ちキューＡ～Ｃを参照し、決められたルールに従って各タスクを起動するスケジューリング手段１、タスク管理テーブルＤに各タスクの状態を格納し、また処理待ちキューＡ～Ｃに処理待ち状態を格納するタスク管理手段２、メモリ、装置などのハードウェア資源や、排他制御のためのフラグ領域などのソフト資源を資源管理テーブルＥを用いて管理する資源管理手段３、およびタスクインタフェース４などがオペレーティングシステムに含まれる部分で、これらにより各優先度のタスク５～７の実行順序が制御される。

【００１２】図１は、本発明の制御方法の一実施例を示すフローチャートで、図３のシステム上で実行される。タスク５～７には、優先度の設定を行うことができるシステムコール（オペレーティングシステムへのサービス要求を行うこと）setprio と、タスク管理手段２を介して優先度の参照を行うことができるシステムコール getprio と、資源管理手段３により排他制御のためのロックの資源（セマフォと呼ぶ）を作成するシステムコール getsem 、セマフォを操作する（ロック／開放する）ためのシステムコール　opsem 、およびセマフォの情報（おもに先にロックしているタスクの番号）を得るためのシステムコール ctrlsem などが用意されていて、次に示すようなアルゴリズムで利用される。

【００１３】まず図１の資源利用サブルーチンでは、最初に資源に対応したセマフォを操作してウエイトなしのロックを試みる（ステップ１０１）。即ち、ウエイトなしを指定して opsem を呼び出すと、セマフォをロックできたか、出来ないかに係わらずすぐにリターンして来る。ここで指定するセマフォとは、予め排他制御する資源に対応させて、ロックの資源を getsem によって作成しておいたものである。

【００１４】リターンの値を参照して（ステップ１０２）、既に他者がロック中であれば、そのロック中のタスクの番号を ctrlsem で得る（ステップ１０３）。又ロック中でなければ、資源操作サブルーチンを呼び出す（ステップ１１０）。ロック中タスク番号を得たときは、そのタスク番号を指定して getprio を呼び出し、ロック中のタスクの優先度を得る（ステップ１０４）。

【００１５】次に、ここで得たロック中のタスクの優先度と自分の優先度とを比較し（ステップ１０５）、ロック中のタスクの優先度の方が高ければ、セマフォを操作してウエイト指定でロックを要求する（ステップ１０９）。ウエイト指定でロックを要求すると、他者がロックを開放するまで処理の実行はブロックされる。又ロック中のタスクの優先度が自分の優先度よりも低ければ、setprio を呼び出してロック中のタスクの優先度を自分の優先度で置き換える（優先度の継承、ステップ１０６）。こうしておいてセマフォを操作し、ウエイト指定でロックを要求する（ステップ１０７）。

【００１６】こうしてウエイト指定でロックを要求すると、他者がロックを開放するまで、処理の実行はブロックされる。そしてタスクが再度スケジュールされたら、先にロックしていたタスクは、セマフォを開放したことを示すので、先ほど与えた自分の優先度を元の優先度に setprio を呼び出して戻してやる（ステップ１０８）。その後、資源操作ルーチンを呼び出し、実際の資源への操作を行う（ステップ１１０）。資源への操作が終了すると、セマフォを開放して（ステップ１１１）終了する。

【００１７】同じ資源を利用するタスクの間に優先度の差異がない場合や、同じ資源を利用する高位の優先度のタスクの優先度と低位の優先度のタスクの優先度の間の優先度をもつタスクが存在しない場合などの、優先度継承の機構がいらない場合には、図２に示したような単純な機構で資源を利用すればよい。

【００１８】

【発明の効果】本発明によれば、優先度の逆転現象によって不確実となっていたリアルタイム性が保証できるようになる。また、ARTS やReal-Time Mach での、オペレーティングシステム側でタスクの優先度の構成に係わらず、全ての資源の排他制御に於て、タスクが排他資源を要求する時点全てで優先度継承のための機構が動作してしまう方式では、常に５～８マイクロ秒のオーバヘッドが発生していたが、本方式によればこれを防止できる。

【図面の簡単な説明】

5

【図１】本発明の優先度制御方法の一実施例を示すフローチャートである。

【図２】優先度継承の必要がないときの優先度制御方法を示すフローチャートである。

【図３】タスクの優先度に基づいて優先度制御を行うシステムの例を示すブロック図である。

【図４】優先度逆転現象の説明図である。

6

【符号の説明】

1　スケジューリング手段

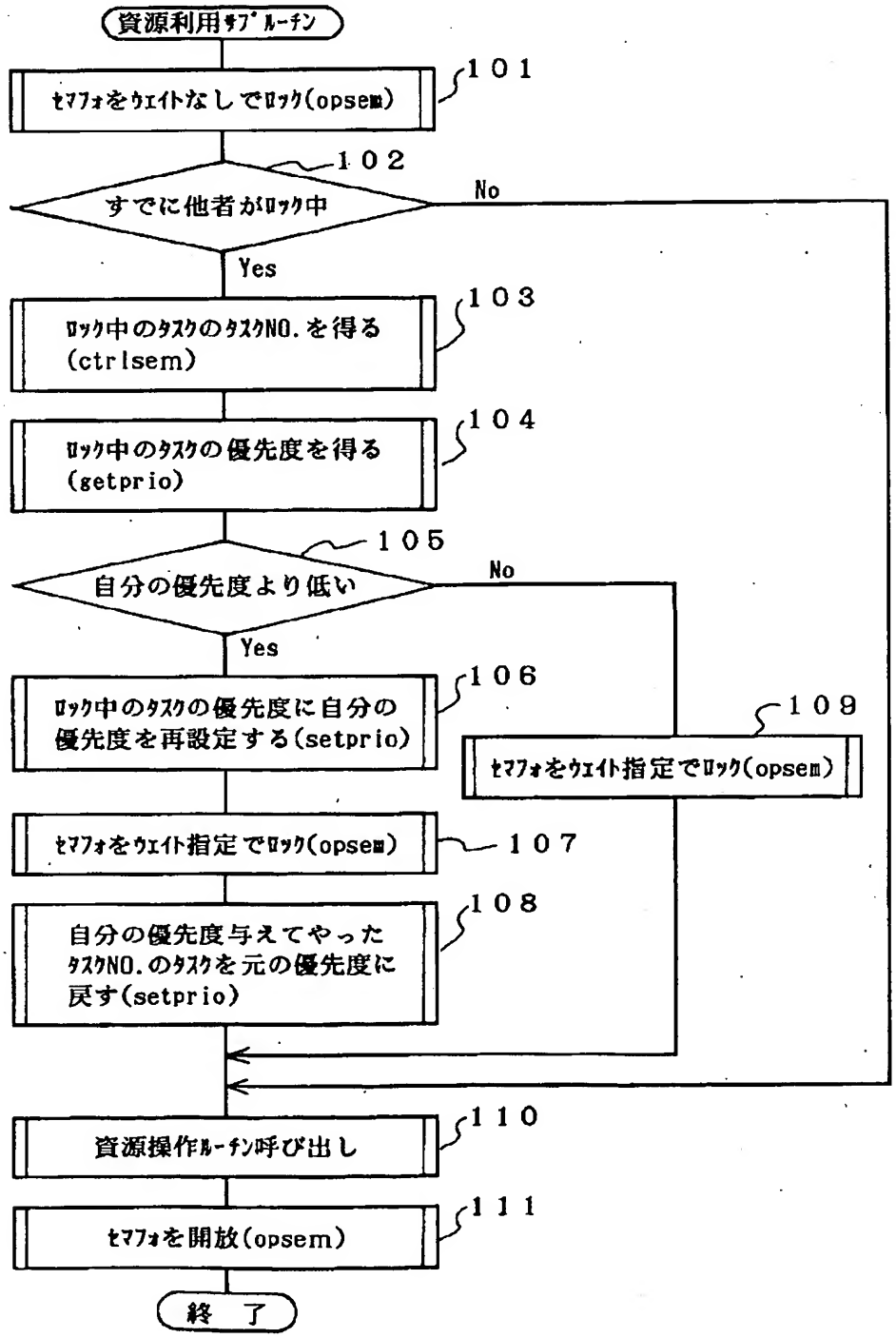
2　タスク管理手段

3　資源管理手段

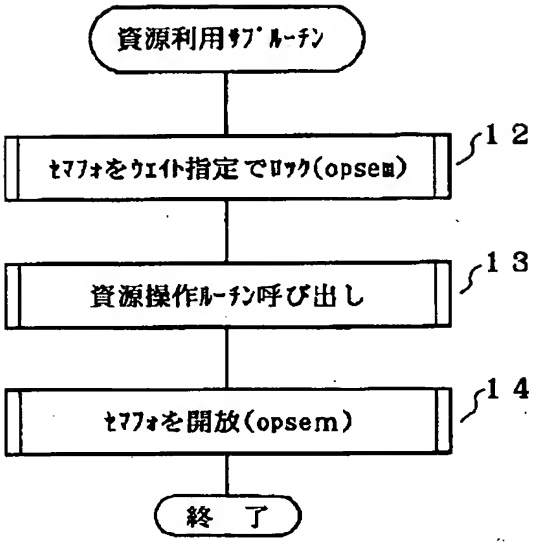
4　タスクインタフェース

5、6、7　タスク

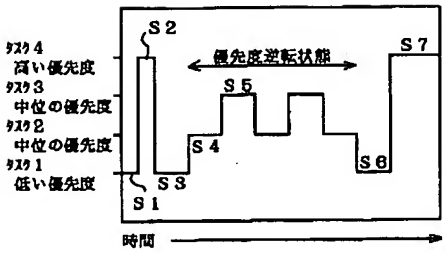
【図１】



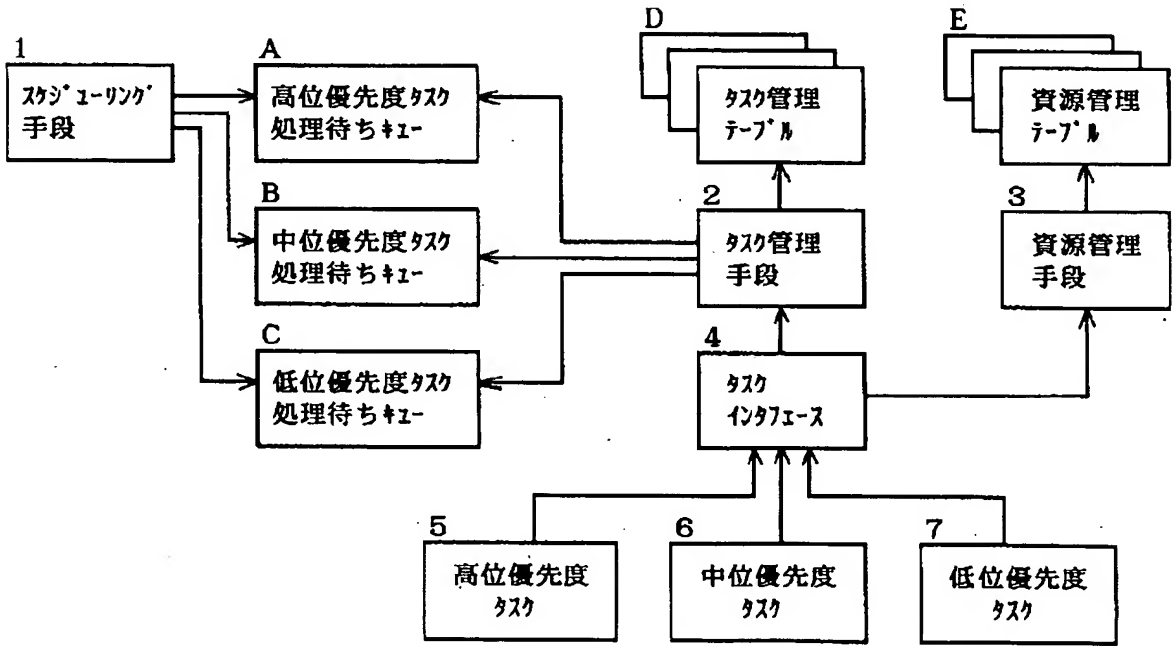
【図２】


資源利用ｻﾌﾞﾙｰﾁﾝ
ｾﾏﾌｫをｳｪｲﾄ指定でﾛｯｸ(opsem)
12
資源操作ﾙｰﾁﾝ呼び出し
13
ｾﾏﾌｫを開放(opsem)
14
終　了

【図４】


ﾀｽｸ4
高い優先度
ﾀｽｸ3
中位の優先度
ﾀｽｸ2
中位の優先度
ﾀｽｸ1
低い優先度
S1
S2
S3
S4
S5
S6
S7
優先度逆転状態
時間

【図３】


1
ｽｹｼﾞｭｰﾘﾝｸﾞ
手段
A
高位優先度ﾀｽｸ
処理待ちｷｭｰ
B
中位優先度ﾀｽｸ
処理待ちｷｭｰ
C
低位優先度ﾀｽｸ
処理待ちｷｭｰ
D
ﾀｽｸ管理
ﾃｰﾌﾞﾙ
E
資源管理
ﾃｰﾌﾞﾙ
2
ﾀｽｸ管理
手段
3
資源管理
手段
4
ﾀｽｸ
ｲﾝﾀﾌｪｰｽ
5
高位優先度
ﾀｽｸ
6
中位優先度
ﾀｽｸ
7
低位優先度
ﾀｽｸ